

Conception et architectures des réseaux

Promotion 14

Travail Pratique

Linux en réseau

A l'issue de ce TP vous devez:

- connaître et savoir utiliser les commandes de base de Unix pour configurer et tester les connexions réseau;
- savoir configurer en réseau un poste de travail sous Linux sans faire recours aux outils graphiques;
- savoir analyser les protocoles de communication à l'aide des programmes pour la capture des trames;
- être capable de concevoir des petites applications en réseau IP en utilisant la programmation par sockets.

Le TP est composé de deux parties (l'étude des outils réseau sous Linux et la programmation d'une application en réseau), vous allez faire des rapports différents pour chaque partie et les remettre à des dates différentes.

Partie 1. Outils réseau sous Linux

1.1 Outils de configuration

Linux contient de nombreux utilitaires permettant de faciliter la configuration et l'administration des interfaces réseau.

Interface réseau: l'interface réseau est représentée physiquement par votre carte réseau mais le terme interface réseau est aussi utilisé pour désigner un nom logiciel auquel assigner une adresse IP (eth0 par exemple). Une adresse IP est toujours assignée à une interface réseau, jamais à un ordinateur. La commande *ifconfig* sert à afficher la configuration des différentes interfaces réseau actives et, en même temps, de les configurer (à faire attention au fait que la configuration sera perdue lors du redémarrage de la machine).

ifconfig : utilitaire standard UNIX permettant d'obtenir des informations sur la configuration de l'interface réseau (carte Ethernet par exemple) :

```
$ ifconfig -a
```

Servez-vous de **man ifconfig** pour connaître les options de cette commande.

Les interfaces réseau d'Ubuntu sont marquées d'habitude avec *eth0*, *eth1*, etc... pour le réseau filaire et avec *wlan0*, *wlan1*, etc. pour le réseau wifi.

route : modifier le table de routage

ping : l'outil le plus simple et le plus pratique des outils réseaux. Ping permet de vérifier si un nom d'hôte distant ou une adresse IP est accessible.

traceroute : utilitaire pour diagnostiquer des problèmes réseaux, en particulier si la commande ping ne réussit pas à atteindre le serveur distant. Il existe des outils avec des fonctionnalités de traceroute et avec des interfaces utilisateur plus évoluées (ex: ***mtr***) ou même graphiques (xtraceroute, par exemple) permettant de visualiser le chemin parcouru par les données entre le client et le serveur.

netstat : utilitaire pour afficher la configuration réseau sous les systèmes UNIX.

dig, **host** et **nslookup** : outils pour interroger les serveurs de nom (résolution de nom).

telnet – client pour accéder à distance à une machine UNIX avec une connexion non-sécurisée: le mot de passe et les données sont transférées en clair.

ftp - client pour télécharger des fichiers avec une connexion non-sécurisée.

Telnet et ftp sont des applications qui utilisent une communication en clair (c'est à dire sans aucune protection de données), il existe des outils sécurisés équivalents qui assurent les mêmes fonctionnalités (vous êtes fortement encouragés de les utiliser pour garantir la sécurité de vos communications) et éviter en même temps d'utiliser des outils non-sécurisés.

ssh - client pour accéder à distance à une machine avec une connexion sécurisée: tout échange de données est cryptée.

scp et **sftp** – utilitaires pour échanger des fichiers avec une machine distante à travers une connexion sécurisée.

Pour plus d'information sur les utilitaires, leur mode d'utilisation et les options correspondantes a utiliser les pages de manuel:

man <nom_utilitaire> ou
info <nom_utilitaire>.

Fichiers de configuration :

/etc/hosts: ce fichier spécifie comment résoudre les noms des machines du réseau local (inutile de mettre en œuvre un serveur DNS pour un petit réseau local). La syntaxe des lignes de ce fichier est:

Adresse IP	Nom de l'hôte	Alias
------------	---------------	-------

Exemple :

127.0.0.1	localhost	
192.168.0.1	sirius.ifi.edu.vn	sirius

/etc/resolv.conf: ce fichier spécifie où résoudre ce qui ne se trouve pas dans /etc/hosts. C'est dans ce fichier que vous devez spécifier les adresses IP des serveurs DNS utilisés pour accéder à Internet (ainsi que des certaines options pour la résolution de nom) en suivant la syntaxe (minimale) suivante:

```
nameserver 192.168.102.1
```

Pour plus d'information sur le format et les options disponibles dans ce fichier, faites un

man resolv.conf

/etc/hostname: ce fichier configure le nom de la machine locale. Au démarrage du système, ce fichier est lu et son contenu est envoyé à la commande hostname. Vous pouvez utiliser la commande hostname pour changer le nom de votre machine.

/etc/network/interfaces: (pour Debian et Ubuntu) configuration des interfaces réseau. Pour les autres distributions Linux il faut regarder la documentation.

Travail à faire:

Découvrez les utilitaires, leur mode d'utilisation et les options correspondantes, analysez les fichiers de configuration de votre machine. (Attention: il ne s'agit pas de citer les outils graphiques dont dispose Gnome ou KDE mais les commandes Linux et les fichiers de configuration!). Certaines applications (paquets) ne sont pas installées par défaut, vous pouvez le faire en ligne de commande (sous Debian et Ubuntu):

aptitude install <nom paquet>

La plupart des distributions actuelles de Linux proposent des interfaces graphiques pour l'installation des paquets, je vous laisse les découvrir par vous mêmes.

Testez chacune de ces commandes les unes après les autres et donnez :

- la liste des interfaces sur votre machine ;
- l'adresse IP de votre machine ;
- l'adresse MAC de votre carte réseau ;
- l'adresse et le masque de votre réseau ;
- la table de routage de votre machine ;
- le nom de la machine d'adresse IP 192.168.100.7, son domaine et le serveur de nom de son domaine ;
- donner la liste des routeurs par lesquels passent les datagrammes entre vous et la machine 192.168.100.7

A découvrir par soi même comment configurer une interface wifi sous Linux sans interface graphique.

1.2 Outils pour la capture des trames

Les outils pour la capture et l'analyse des trames sont indispensables pour tout développement de protocole. La capture de trames permet d'analyser le bon fonctionnement des différents protocoles ainsi que les différents temps d'exécution de ces protocoles. Vous trouverez en annexe un brève rappel sur la structure des unités des données dans les réseaux TCP/IP. Pour plus d'information a regarder les RFC correspondants.

Linux dispose de l'utilitaire *tcpdump* qui permet de capturer des trames en mode textuel et *Wireshark* (également connu jusqu'en 2006 comme *Ethereal*) - un logiciel libre, assez puissant (disponible sur un grand nombre de plates-formes matérielles et de systèmes d'exploitation) pour capturer et analyser les trames en mode graphique. Vous pouvez télécharger la dernière version de ce logiciel sur www.wireshark.org, il est également disponible dans les dépôts des distributions Linux sous forme précompilée. Vous trouverez une brève description du mode d'utilisation de ce logiciel en suivant les liens sur la page de cours.

Travail à faire:

Analyse du protocole de résolution d'adresse ARP

Démarrez une capture en ayant préalablement paramétré vos filtres afin de ne visualiser que les trames qui vous intéressent. Consultez le cache ARP de votre machine (commande `arp -a`). Pinguez une adresse IP de votre réseau local qui ne soit pas dans le cache ARP de votre machine. En analysant les trames capturées, déterminez le fonctionnement du protocole ARP. Pour ce faire, consultez en détail la couche liaison (Ethernet) des trames et comparez en détail le contenu de la trame au niveau ARP/RARP entre le *Reply* et le *Request*.

Analyse du protocole Traceroute

De même, déterminez le fonctionnement du protocole *traceroute*. Pour ce faire, effectuez un *traceroute* (vous pouvez utiliser un autre logiciel équivalent, par exemple *tcptraceroute*, *tracpath*, etc.) vers le hôte 192.168.100.2. Recherchez au niveau IP, les champs qui varient entre les triplets successifs d'écho (excepté le champ identification et checksum qui ne sont d'aucun intérêt). Que remarquez-vous sur la succession des adresses IP affichées par rapport à l'ensemble des interfaces réseaux réellement traversées pour joindre la destination (Aide: en étudiant la topologie du réseau et plus particulièrement le plan d'adressage, déterminez la succession des interfaces traversées par vos paquets).

Analyse du protocole telnet et la capture des informations qui circulent en clair sur le réseau

Lancez le serveur telnet sur une des machines de votre groupe. Ajoutez un utilisateur avec le nom *test*. Faites un telnet sur cette machine. Arrêtez la capture et affichez le résultat. A quoi correspondent les adresses MAC source et destination au niveau liaison (Ethernet)? Expliquez votre observation. Faites un telnet sur un ordinateur de votre réseau pour vous connecter sur la machine distante. Évitez les fautes de frappes qui génèrent du trafic inutile. Arrêtez et affichez la capture. Étant donné le nombre important des trames capturées, lors de l'analyse des trames prenez seulement en compte les trames dont les segments TCP encapsulent des données. Vous pouvez évidemment, vous connecter sur le serveur telnet de jupiter pour faire les mêmes manipulations.

- Indiquez les ports TCP source et destination utilisés par telnet (convertir le format hexadécimal en base 10);
- Retrouvez les mots de passe que vous avez utilisé avec telnet? Où cherchez-vous ces informations?

Question: Lorsque vous êtes sur un réseau Ethernet, capturez-vous tous les trames qui traversent le réseau? Expliquez. Idem pour un réseau wifi.

Analyse du protocole TCP

On vous propose de faire une analyse détaillée du protocole TCP par capture de trames. Vous allez télécharger sur votre machine un fichier avec un protocole qui utilise une connexion TCP et capturer en même temps tous les paquets qui concernent cette communication. On utilisera le protocole http pour télécharger sur votre machine le fichier `index.html` à partir du serveur `intranet.dorsale.ifi`:

- Lancer la capture avec `tcpdump` dans un fichier (pensez à filtrer les paquets capturés pour avoir que les informations qui concernent la connexion correspondante). Donnez la commande que vous avez utilisée.
- Lancer la commande `wget http://intranet.dorsale.ifi/index.html`
- Arrêtez la capture et analysez les résultats:
 - Donnez le listing de la capture dans le rapport et identifiez les zones qui correspondent aux phases de connexion, transfert de données, déconnexion;
 - Identifiez les numéros initiaux de séquence (Initial Sequence Number - ISN) utilisés dans les deux sens;
 - Représentez sur un schéma cette échange en montrant l'évolution des numéros de séquence, de la fenêtre, des acquittements, des fanions, etc.

Vous devez remettre un petit rapport de groupe pour cette partie qui sera discuté avec vous pendant une des séances suivantes.

Le contenu du rapport pour cette partie

- Les informations sur les interfaces de votre machine demandées au début de cette partie;
- Explications sur la configuration des interfaces wifi sous Linux;
- Une brève analyse des protocoles *ARP*, *traceroute* et *telnet* avec les captures des trames correspondantes (les parties les plus concluantes, évidemment). On vous demande une vérification pratique des protocoles à l'aide des trames capturées même si vous connaissez en théorie leur fonctionnement;
- Une analyse détaillée du protocole TCP (capture; analyse, diagramme, etc.)
- Vos conclusions.

Les rapports doivent être envoyés par courrier électronique avec le sujet "CAR, TP1, équipe x" à l'adresse victor.moraru@auf.org¹, la date limite de remise des rapports est le 25 mars 2009 avant 09h00

¹ Lorsque vous envoyez votre rapport par courrier électronique demandez un accusé de réception (dans Thunderbird: menu "Options" --> "Demander un accusé de réception"). Ça me permettra d'éviter l'envoi manuel des réponses à chaque message. Merci.

Conception d'un client FTP

2.1 Objectif

Il s'agit d'un projet réalisé en équipe sur la conception d'un client minimal FTP basé sur le RFC du protocole FTP. Le langage de programmation conseillé est C/C++.

2.2 File Transfer Protocol

Le protocole de transfert de fichier est défini dans la RFC 959 [1]. Étudiez ce document et essayez de comprendre les fonctionnalités FTP en termes des commandes USER, PASS, SYST, PASV, LIST, RETR, STOR et QUIT.

2.3 Énoncé des travaux

L'idée de base de votre projet est la conception et le codage d'un client FTP en vous basant sur RFC959 et avec un jeu restreint des commandes (voir le tableau).

Le programme devrait être exécuté avec *ftp [nom_hôte]*. Il ouvre une session sur le serveur distant (par défaut sur le port 21) en utilisant le nom d'hôte ou son adresse IP et effectue les transferts de fichiers en mode passif. Si aucun argument n'est fourni, le programme client ne crée aucune connexion jusqu'à ce qu'il reçoive une commande (voir ci-dessous).

L'interface du client FTP	Commandes RFC959 correspondantes	Comportement
open [nom, IP] [port]	-	Établir une connexion TCP avec le serveur.
name: <nom>	USER <nom>	Envoyer l'identificateur de l'utilisateur au serveur
password: <mot_de_passe>	PASS <mot_de_passe> SYST	Envoyer au serveur le mot de passe de l'utilisateur
cd <répertoire>	CWD <répertoire>	Changer le répertoire de travail courant pour le sous-dossier <répertoire>
ls	PASV LIST	Demander au serveur de renvoyer le contenu de son répertoire courant
get file	PASV RETR file	Obtenir un fichier à partir du répertoire distant courant.
put file	PASV STOR file	Stocker un fichier dans le répertoire distant courant.
close	QUIT	Fermeture de la connexion sans quitter ftp
quit	QUIT (si ouvert)	Fermeture de la connexion et/ou quitter ftp

Notez que le mode passif (énoncé par la commande PASV) demande au serveur de renvoyer son adresse IP et le port sur lequel il écoute pour les données de la part du client.

Pour tester votre programme client, vous devez effectuer la vérification suivante: lancer votre logiciel et connectez-vous sur jupiter.dorsale.ifi, le port 21. Identifiez-vous avec votre login ou avec un login et un mot de passe communs qui vous seront fournis en classe. Testez les commandes **cd**, **ls**, **get**, **put**, **close** et **quit**. Téléchargez sur votre machine un des fichiers des dossiers de ftp. Mettez sur le serveur un de vos fichiers.

Utilisez telnet pour ouvrir une session en direct sur le serveur ftp: **telnet jupiter.dorsale.ifi 21** ou sur une autre machine. Cette opération va vous aider à comprendre ce que le serveur vous envoie en réponse à chaque commande que vous avez tapé. Exécution du client ftp d'Unix pourrait vous aider à justifier votre application en termes de performances et des fonctionnalités d'exécution.

Vers la fin de vos expériences réalisez une capture de trames entre votre client et le serveur et

prouvez le fonctionnement en mode passif, ainsi que la conformité des échanges à la RFC959.

2.4 Barème de correction²

Votre projet final sera évalué et noté selon les critères suivants:

Exposé du problème et documentation

2 points

Donnez une brève description de votre projet, ses limitations (le degré de conformité à la RFC 959) et les instructions nécessaires pour le compiler et l'utiliser sous forme d'une petite documentation

Discussion

6 points

Description détaillée de votre programme: efficacité, difficultés (particularités) rencontrées, limites, algorithmes, organigrammes, etc. ... Exactitude des opérations, l'interface utilisateur, etc.

Évaluation de performance

3 points

Comparez les durées d'exécution des commandes GET et PUT sur les mêmes fichiers de votre client ftp et du programme ftp de Unix. Donnez les détails de vos expériences et les résultats de mesure.

Perspectives

1 point

Discussions sur des éventuelles extensions fonctionnelles de votre propre programme et/ou du client ftp d'Unix (donnez vos propres réflexions !), analyse de la différence de performance entre votre propre programme et du client ftp d'Unix.

Conclusions et recommandations

3 points

Évaluation des résultats, explications, améliorations, recommandations, atteinte des objectifs, etc. ...

Qualité du logiciel

3 points

Code commenté, bien structuré, etc. ...

Présentation et qualité du français

2 points

Des crédits supplémentaires allant jusqu'à 2 points pourront être attribués si vous mettez en œuvre des commandes ftp supplémentaires ou des nouvelles fonctionnalités intéressantes et utiles.

Le contenu du rapport pour cette partie

- ✓ La présentation du protocole et de votre client (le scénario, les algorithmes, les formats, les types des messages, etc.)
- ✓ Le code documenté de votre programme, le mode d'emploi, les résultats obtenus.
- ✓ Vos conclusions.

Vous devez remettre votre rapport par courrier électronique à l'adresse victor.moraru@auf.org avec le sujet "CAR, Projet 1, équipe x" en fichier attaché (format pdf) et une archive avec votre code source, avec des fichiers complémentaires nécessaires pour compiler (le makefile) et lancer votre application et avec des instructions bien précises sur l'utilisation de votre logiciel.

La date limite de remise des rapports est le 14 avril 2009 avant 12h00.

Références

1. Request for Comments 959 "File Transfer Protocol (ftp)" <http://www.ietf.org/rfc/rfc0959.txt>
la traduction française: <http://abcdrfc.free.fr/rfc-vf/rfc959.html>

² Les poids sont à titre indicatif

Structure des unités de données

Structure de la trame Ethernet

```

+---6-octets---+---6-octets---+2o---+ - - - - - +
| adresse      | adresse      | type | données |
| destination  | source      |     |         |
+-----+-----+-----+-----+

```

Trame présentée sans préambule ni CRC.

Quelques types : 0x0200 = XEROX PUP 0x0800 = IPv4
 0x0806 = ARP 0x8035 = RARP
 0x814C = SNMP 0x86DD = IPv6
 <1500 - 802.3 ...

La liste complète des types des trames Ethernet:

<http://www.iana.org/assignments/ethernet-numbers>

Structure du paquet IPv4

```

<-----32bits----->
<-4b->      <--8bits--><-----16bits----->
+-----+-----+-----+-----+
| Ver | IHL |    TOS    | Longueur totale |
+-----+-----+-----+-----+
| Identificateur | Fl |    FO    |
+-----+-----+-----+-----+
| TTL    | Protocole | Somme de ctrl (entete) |
+-----+-----+-----+-----+
| Adresse Source |
+-----+-----+-----+-----+
| Adresse Destination |
+-----+-----+-----+-----+
... Options
+-----+-----+-----+-----+
... Données
+-----+-----+-----+-----+

```

Ver = Version d'IP,

IHL = Longueur de l'entête IP (en mots de 32 bits),

TOS = Type de service,

Longueur totale du paquet IP (en octets).

Fl (3 bits) = Indicateurs pour la fragmentation

1er = réservé,

2ème = ne pas fragmenter et

3ème = fragment suivant existe),

FO (13 bits) = décalage du fragment (valeur a multiplier par 8 octets),

TTL = durée de vie restante.

Quelques protocoles transportés :

1 = ICMP	8 = EGP
2 = IGMP	11 = DoP
4 = IP (encapsulation)	17 = UDP
5 = Stream	36 = XTP
6 = TCP	46 = RSVP

Structure du datagramme ICMP

```

<-----32bits----->
+-----+-----+-----+-----+
| Type    | Code    | Somme de contrôle (dtg) |
+-----+-----+-----+-----+
|    Variable (généralement non utilisé)    |
+-----+-----+-----+-----+
...    Datagramme original + 8 octets
+-----+-----+-----+-----+

```

Quelques types ICMP : 0 = Demande d'écho,
 8 = Réponse d'écho,
 11 = Durée de vie écoulée,
 12 = Erreur de paramètre...

Structure ARP

```

+16b--+16b--+8b+8b+16b+---lgHW--+lgP+---lgHW--+lgP--+
| type | type | lg | lg | Op | Emetteur | Emt. | Recept. | Rcpt |
| HW   | Proto | HW | P   |    | adr. HW | adrP | adr. HW | adrP |
+-----+-----+-----+-----+-----+

```

Quelques types : 0x0001 = Ethernet 0x0800 = DoD Internet

Opérations : 0x0001 = Requête 0x0002 = Réponse

Structure du datagramme UDP

```

<-----32bits----->
+-----+-----+-----+-----+
| Port Source            | Port Destination    |
+-----+-----+-----+-----+
| Longueur UDP           | Somme de ctrl (message) |
+-----+-----+-----+-----+
... Données
+-----+-----+-----+-----+

```

Structure du segment TCP

```

<-----32bits----->
<-4b->      <-6bits-><-----16bits----->
+-----+-----+-----+-----+
| Port Source            | Port Destination    |
+-----+-----+-----+-----+
| Numéro de Séquence    |
+-----+-----+-----+-----+
| Numéro d'Acquittement |
+-----+-----+-----+-----+
| THL |            | Flag    | Taille Fenetre |
+-----+-----+-----+-----+
| Somme de ctrl (message) | Pointeur d'urgence |
+-----+-----+-----+-----+
... Options
+-----+-----+-----+-----+
... Données
+-----+-----+-----+-----+

```

THL = longueur de l'entête TCP sur 4 bits (en mots de 32 bits)

Flags (6 bits) = indicateurs (du bit de poids le plus faible au plus fort):

* 0 = Fin	(FIN)
* 1 = Synchronisation	(SYN)
* 2 = Réinitialisation	(RST)
* 3 = Données immédiates	(PSH)
* 4 = Acquittement	(ACK)
* 5 = Données urgentes	(URG)

Options = suites d'option codées sur

* 1 octet à 00 = Fin des options

* 1 octet à 01 = NOP (pas d'opération)

* plusieurs octets de type TLV (Type, Longueur, Valeur)

T = un octet de type:

2 Négociation de la taille max. du segment

3 Adaptation de la taille de la fenêtre

4 Autorisation des acquittements sélectifs

8 Estampilles temporelles

L = un octet pour la taille totale de l'option

V = valeur de l'option (sur L-2 octets)

Services associés aux ports

ftp-data	20/tcp	ssh	22/udp
ftp	21/tcp	domain	53/udp
ssh	22/tcp	tftp	69/udp
telnet	23/tcp	www	80/ucp
smtp	25/tcp	pop-3	110/udp
domain	53/tcp	snmp	161/udp
www	80/tcp	snmp-trap	162/udp
imap	143/tcp	imap	143/udp
pop-3	110/tcp	pop-3	110/udp
ntp.....	123/tcp	ntp	123/udp

La liste complète des numéros des ports:

<http://www.iana.org/assignments/port-numbers>